

1/11

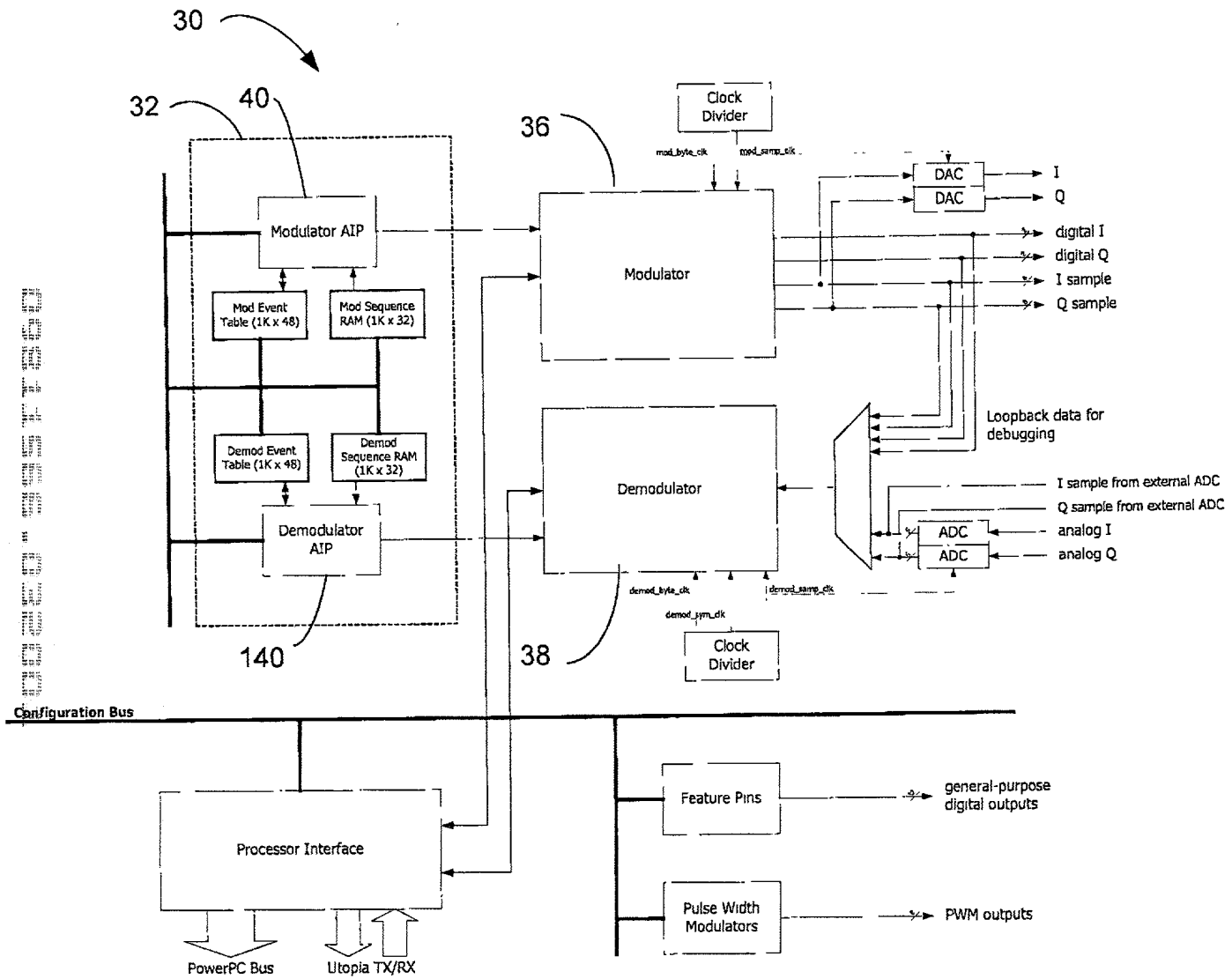


Figure 1

2/11

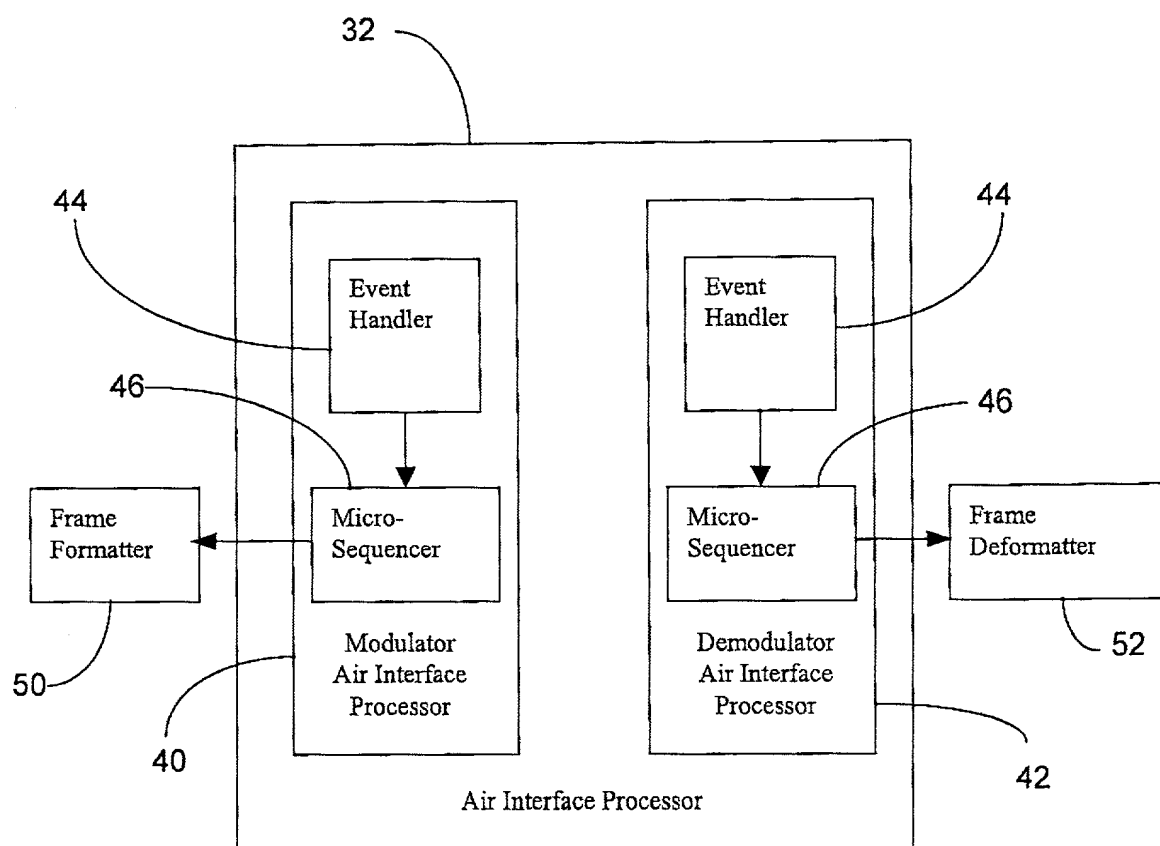


Figure 2

3/11

| | | | | | | | | | |
|-------------|--|--|-------------------------|-----|--------------|-------------|--------------|--------------|----|
| Instruction | 4444444433333333333322222222111111111111 | 765432109876543210987654321098765432109876543210 | | | | | | | |
| Type 1 | 00 | ALU opcode | SW Rd | Rn | operand 2 | branch code | pass address | fail address | |
| Type 2 | 0100 | Rhi | - | Rlo | 32-bit data | | | | |
| Type 3 | 0101 | Rhi | Imm_lo | | 32-bit data | | | | |
| Type 4 | 0110 | Rhi | - | Rlo | - | | | | Rd |
| Type 5 | 0111 | Rhi | Imm_lo | | - | | | | Rd |
| Type 6 | 1TDQ | 0 | Microsequence r address | | trigger time | | | | |
| Type 7 | 1 | - | 01 | - | burst info | | | | |
| Type 8 | 1 | - | A11 | - | | | | mask | |

Instruction Type 1: ALU Operations

Instruction Type 2: Write register

Instruction Type 3: Write register immediate

Instruction Type 4: Read register

Instruction Type 5: Read register immediate

Instruction Type 6: Trigger

Instruction Type 7: BURST

Instruction Type 8: WAIT

[A='0' → until any of (R12 and mask) bits are set]

[A='1' → until all of (R12 and mask) bits are set]

Figure 3: Event Handler Instruction Set Summary

4/11

[illegible]

Figure 4: Register Access Instructions

[illegible]

Figure 5: Data Scheduling Instructions

[illegible]

Figure 6: Burst Descriptor Instruction

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----|----|-----------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|
| 33 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | |
| PS | | value to DDS/Fractional-N counter | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 7: Modulator Burst Info Field Format

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|
| 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| User ID | | | | | | | | | | | | | | | PS | | Expected Length | | | | | | | | | | | | | | |

Figure 8: Demodulator Burst Info Field Format

[illegible]

Figure 9: Processor Wait Instruction

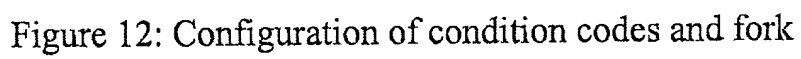
5/11

| opcode | name | Description |
|--------|------|---|
| 00000 | JZ | Jump to Zero |
| 00001 | CJS | Conditional Jump to Subroutine |
| 00010 | JMAP | Jump Map |
| 00011 | CJP | Conditional Jump Pipeline |
| 00100 | PUSH | Push/Conditional Load Counter |
| 00101 | JSRP | Conditional Jump to Subroutine |
| 00110 | CJV | Conditional Jump Vector |
| 00111 | JRP | Conditional Jump |
| 01000 | RFCT | Repeat Loop Counter Not Equal to Zero |
| 01001 | RPCT | Repeat Pipeline Counter Not Equal to Zero |
| 01010 | CRTN | Conditional Return |
| 01011 | CJPP | Conditional Jump Pipeline and Pop |
| 01100 | LDCT | Load Counter and Continue |
| 01101 | LOOP | Test End of Loop |
| 01110 | CONT | Continue |
| 01111 | TWB | Three Way Branch |
| 10000 | FORK | Multway Branch |
| others | | reserved |

Figure 10: Microsequencer Instruction Set

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|-------|----|----|----|----|-------|---|---|---|---|---|---|---|----|----|---|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| OPCODE | | | | | | | | EMIT | | | | | | | | CCSEL | | | | CP | FFCMD | | | | | | | | SB | OC | - | SR |

Figure 11: Microsequencer memory format

[illegible]

7/11

| 1 | x | x | 1 | x | Match | |
|---|---|---|---|---|-------|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | |
| 0 | 0 | 0 | 1 | 0 | 0 | |
| 0 | 0 | 0 | 1 | 1 | 0 | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | |
| 0 | 0 | 1 | 1 | 0 | 0 | |
| 0 | 0 | 1 | 1 | 1 | 0 | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | |
| 0 | 1 | 0 | 1 | 0 | 0 | |
| 0 | 1 | 0 | 1 | 1 | 0 | |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | |
| 0 | 1 | 1 | 1 | 0 | 0 | |
| 0 | 1 | 1 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 0 | 0 | C |
| 1 | 0 | 0 | 0 | 1 | 0 | |
| 1 | 0 | 0 | 1 | 0 | 1 | |
| 1 | 0 | 0 | 1 | 1 | 1 | |
| 1 | 0 | 1 | 0 | 0 | 0 | C |
| 1 | 0 | 1 | 0 | 1 | 0 | |
| 1 | 0 | 1 | 1 | 0 | 1 | |
| 1 | 0 | 1 | 1 | 1 | 1 | |
| 1 | 1 | 0 | 0 | 0 | 0 | C |
| 1 | 1 | 0 | 0 | 1 | 0 | |
| 1 | 1 | 0 | 1 | 0 | 1 | |
| 1 | 1 | 0 | 1 | 1 | 1 | |
| 1 | 1 | 1 | 0 | 0 | 0 | C |
| 1 | 1 | 1 | 0 | 1 | 0 | |
| 1 | 1 | 1 | 1 | 0 | 1 | |
| 1 | 1 | 1 | 1 | 1 | 1 | |

Figure 13

8/11

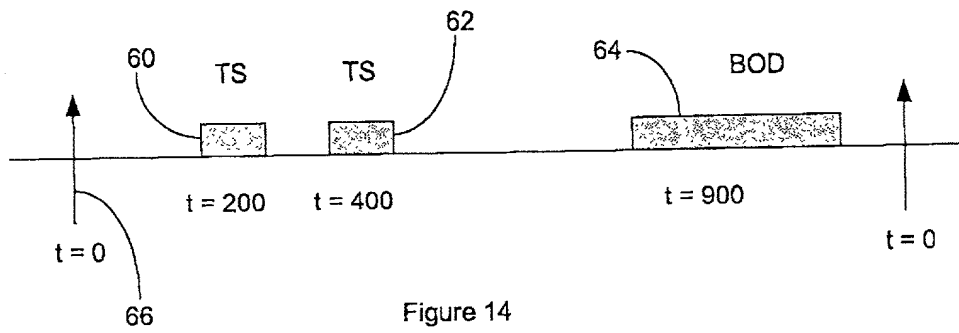


Figure 14

| Event Handler | | Microsequencer | |
|---------------|----------|----------------|------------------|
| Time | Sequence | Sequence | Actions |
| T=200 | TS | TS | send header |
| T=400 | TS | | send 1 ATM cell |
| T=600 | BOD | | add RS encoding |
| | | BOD | send header |
| | | | send 3 ATM cells |
| | | | add RS encoding |

Figure 15

Figure 16

9/11

Terminal Modulator Block Diagram

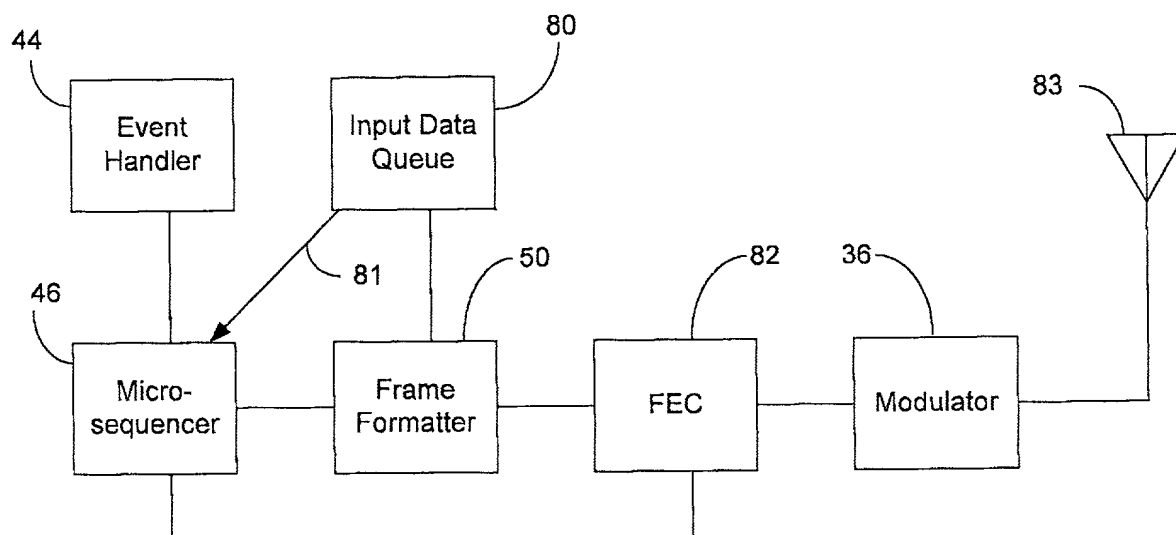


Fig 17

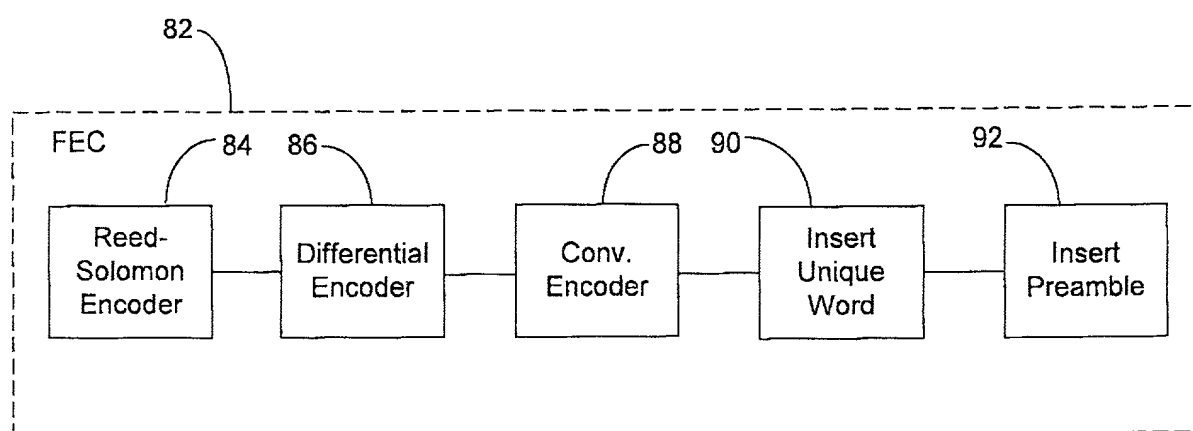


Fig 18

10/11

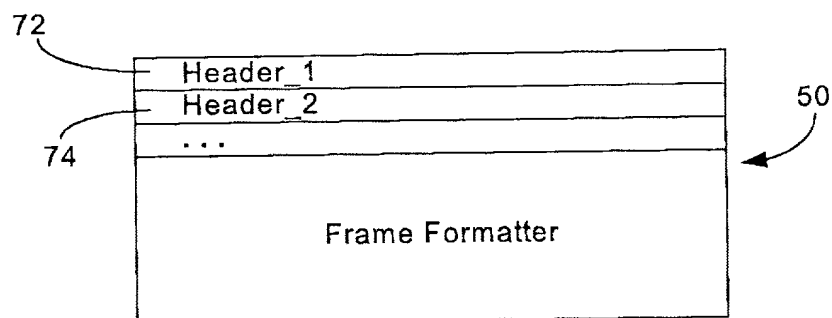


Figure 19

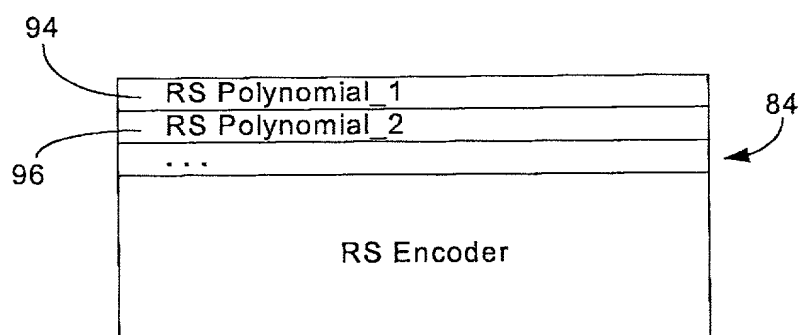


Figure 20

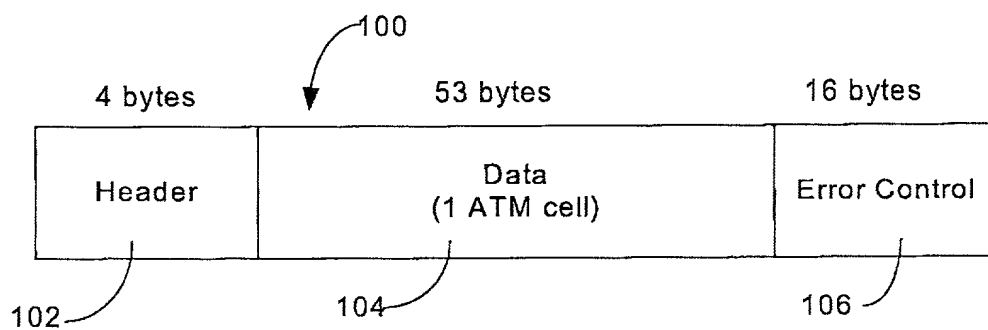


Figure 21

11/11

Terminal Demodulator Block Diagram

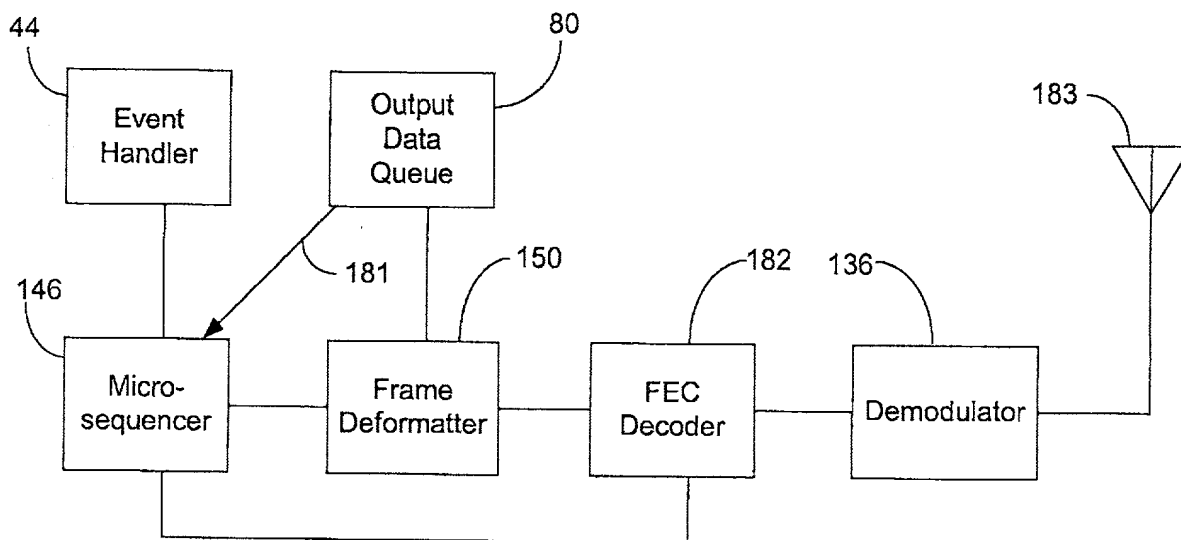


Fig 22

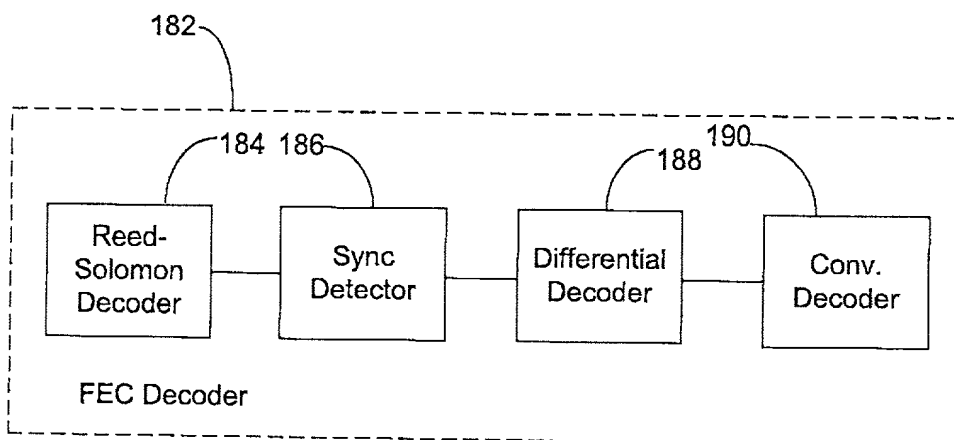


Fig 23